SaprotHub: Making Protein Modeling Accessible to All Biologists

Jin Su[1], Zhikai Li[1], Chenchen Han[1], Yuyang Zhou[1], Yan He[1], Junjie Shan[1], Xibin Zhou[1], Xing Chang[1], Dacheng Ma[2], The OPMC[5], Martin Steinegger[3], Sergey Ovchinnikov[4], Fajie Yuan[1]

[1]Westlake University.
[2]Zhejiang Lab.
[3]Seoul National University.
[4]MIT.
[5]Open Protein Modeling Consortium.

Correspondence to: yuanfajie@westlake.edu.cn;

Abstract

Training and deploying deep learning models pose challenges for users without machine learning (ML) expertise. SaprotHub offers a user-friendly platform that democratizes the process of training, utilizing, storing, and sharing protein ML models, fostering collaboration within the biology community—all achievable with just a few clicks, regardless of ML background. At its core, Saprot is an advanced, foundational protein language model. Through its ColabSaprot framework, it supports potentially hundreds of protein training and prediction applications, enabling the co-construction and co-sharing of these trained models. This enhances user engagement and drives community-wide innovation.

Proteins, the building blocks of life, are essential for a myriad of biological processes and pivotal in medical breakthroughs, pharmaceutical innovations, and genetic discoveries [5, 20, 10]. Despite their importance, understanding protein structure and function has been a considerable challenge in the scientific community. The success of AlphaFold2 [22] in CASP14 marks a new era in structural biology by achieving accuracy of protein structure prediction at the experimental margin of error. Simultaneously, large-scale protein language models (PLMs) are driving substantial advances in protein function prediction.

In this context, several notable PLMs have emerged, including the ESM models [36, 25, 54, 59], ProtTrans [6], ProstT5 [13], UniRep [2], Tranception [31], ProGen [26, 30], and ProtGPT2 [7], each demonstrating remarkable efficacy in their respective tasks. However, training and deploying these machine learning (ML) models for proteins often pose major challenges for researchers without ML expertise. These include selecting appropriate model architectures, managing coding intricacies, preprocessing large datasets, training model parameters, evaluating and interpreting results. This complexity often deters non-ML researchers, hindering their active involvement in this domain, especially as AI models continue to grow more sophisticated.

Given these challenges, ColabFold [29] emerged as a pioneering initiative by deploying AlphaFold2 predictions on Google Colaboratory, making protein folding accessible to researchers of all backgrounds. However, ColabFold primarily emphasizes model inference, lacking support for researchers in the complex tasks such as training their own ML models.

In response to this need, we present SaprotHub, an easy-to-use platform based on Google Colaboratory tailored for protein function-related predictions. SaprotHub empowers researchers by enabling them to create and train their own task-specific models without requiring advanced ML or coding expertise. Additionally, SaprotHub distinguishes itself by supporting a broad range of protein function predictions (Fig. 1c, 2b), rather than being confined to a specific task.

Furthermore, we introduce the *Open Protein Modeling Consortium* (OPMC) (Supplementary) with the vision to create a unified repository for decentralized protein prediction models. Within the OPMC framework, researchers can easily share their individually trained models, fostering both the direct use and collaborative construction of protein models. This approach enables continue learning and improvement of

Partial materials of Saprot was reported in [37], following Nature journal's conference proceedings policy, see here.

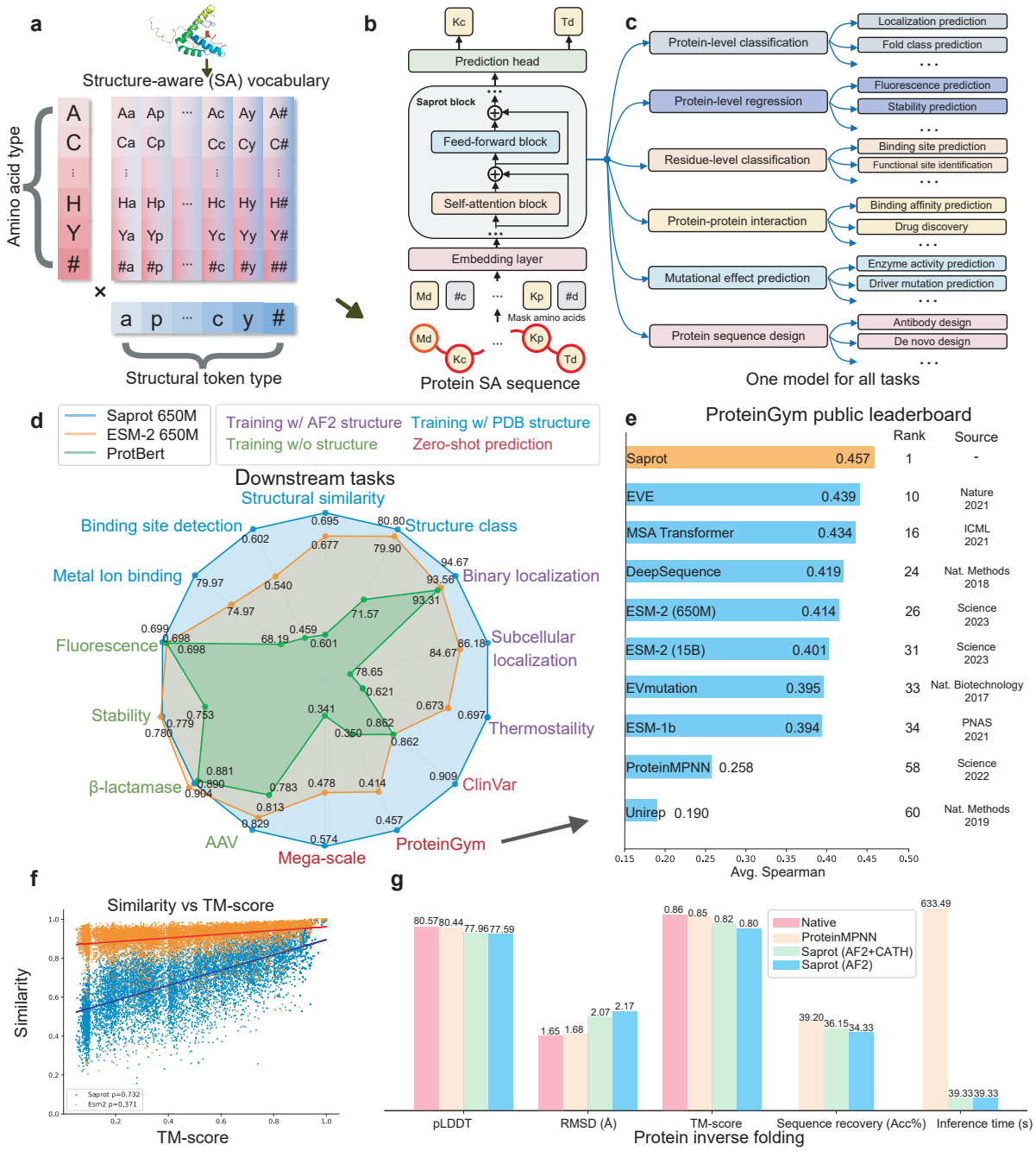


Fig. 1: **Illustration of Saprot. a,** The proposed SA vocabulary. #a, #p,...,#y represent only the 3Di token being visible, while A#, C#,..., Y# represent the AA token being visible. **b,** Architecture and pre-training of Saprot. **c,** Saprot supports numerous protein prediction tasks, see more specific tasks in Supplementary Table 5. **d,** Performance comparison on 14 diverse prediction tasks. **e,** Saprot ranked *first* on ProteinGym leaderboard, outperforming 60 single and hybrid methods. Hybrid methods are not shown here (see Supplementary Fig. 1). **f,** Comparison of Saprot and ESM-2 in structural representation ability. The y-axis represents the cosine similarity score of the last layer representation for every pair of proteins, while the x-axis represents the TM-score of the protein pairs. Saprot shows higher Pearson correlation coefficient (ρ) between x and y axes, indicating more accurate protein structure encoding. **g,** Performance on protein sequence design. Although Saprot was not trained with a typical generation loss, its protein design performance rivals that of ProteinMPNN, with over 16 times faster inference speed. Saprot (AF2+CATH) indicates that it was pre-trained using AlphaFold2 predicted structures and fine-tuned with experimental structures from the CATH dataset. CATH is also used for training ProteinMPNN.

models through collective efforts. SaprotHub, as the first PLM Hub to join OPMC, marks the first step to bring the OPMC vision to fruition.

SaprotHub comprises three novel contributions: a new protein language model called Saprot (Fig. 1a,b), ColabSaprot (Fig. 2a,b,c) for fine-tuning (or re-training) and inference of Saprot using adapter learning techniques [17] on the Colab platform, and a centralized community model repository (Fig. 2d) for saving, sharing, searching, and collaborating on fine-tuned Saprot models. Through the innovative integration of

universal protein language models, Colab-based cloud computing, and adapter-based fine-tuning, SaprotHub addresses several key challenges. These challenges include the difficulty of sharing and aggregating large-scale models, the risk of catastrophic forgetting during continual learning, and the need to protect proprietary data. Below, we describe these three modules in detail.

We developed Saprot, a cutting-edge, large-scale protein language model that forms the foundation for ColabSaprot and SaprotHub. A key feature of Saprot is the introduction of a novel structure-aware (SA) protein representation method (Fig. 1a), departing from classical amino acid (AA) sequences or explicit 3D structures. Specifically, we introduce a unique SA vocabulary (SAV) where $SAV = \{\mathcal{V}, \mathcal{F}\}$, combining 20 amino acids ($\mathcal{V}$) and 20 structural (3Di) alphabet tokens ($\mathcal{F}$), generated by encoding 3D structures with Foldseek [39]. SA tokens encompass all AA and 3Di combinations in SAV, allowing proteins to be represented as SA strings that capture both primary and tertiary structures (Fig. 1b). Despite its conciseness, applying the SA vocabulary successfully addresses the key challenges of scalability and overfitting in training on large-scale *AlphaFold2-generated* atomic structures (see [37, 11, 4]). The elegant use of SA token sequences to represent proteins has gained increasing attention in subsequent studies [9, 24, 33, 11, 19] and holds promise as a new paradigm for protein representation.

We adopt a bidirectional Transformer [47] for Saprot (Fig. 1b). During pre-training, Saprot uses SA sequences instead of AA sequences, *partially* masks random SA tokens, and predicts their complete forms (Methods and Fig. 1b). Unlike models built on existing PLMs, Saprot was trained from scratch with 650 million parameters on over 40 million protein SA sequences, filtered from 214 million AlphaFold database proteins [40] with a 50% identity threshold, over three months using 64 NVIDIA 80G A100 GPUs. This made it the largest structure dataset for PLM training at the time [37]. This huge computational demand mirrors that of ESM-2 (650M) and AlphaFold3 [1].

After pre-training, Saprot has become a foundational protein language model, excelling in various protein prediction tasks (Fig. 1c), including supervised training, zero-shot prediction, and protein sequence design. In supervised training, it handles both protein-level (e.g., fluorescence, thermostability, structure, and folding class prediction) and residue-level (e.g., binding site detection) tasks [18]. In zero-shot scenarios, Saprot predicts enzyme activity [12], viral mutation fitness [14], disease variant effects, and driver mutations [8] without fine-tuning (Methods). Though not designed with an explicit generative loss, Saprot (using masked language modeling loss) shows remarkable proficiency in protein sequence design, achieving a 16× speedup in inference compared to ProteinMPNN [3] (Fig. 1g, 2h, Supplementary Fig. 2).

Fig. 1d shows Saprot's exceptional performance across 14 diverse protein prediction tasks compared to the ESM-2 [25] and ProtBert [6] PLMs. When protein structures are available, Saprot consistently outperforms both models in tasks marked in purple, blue, and red. Without structures available, Saprot remains highly competitive (green tasks). Notably, Saprot with frozen representation substantially outperforms ESM-2 in three zero-shot mutational effect prediction tasks: Mega-scale [38] (0.574 vs. 0.478), ProteinGym [32] (0.457 vs. 0.414), and ClinVar [23] (0.909 vs. 0.862).

Saprot held the top spot on the public blind ProteinGym leaderboard (Fig. 1e, Supplementary Fig. 1) for several months after its release. It surpasses over 60 notable methods, including ESM-2 (650M, 3B, 15B), EVE [8], ESM-1v [54], MSA Transformer [59], ESM-IF [15], DeepSequence [35], and all hybrid models. Notably, the 35M parameter version of Saprot outperforms the 15B parameter version of ESM-2. Supplementary Tables 2-3 and Fig. 1, 3 provide more comparisons. Recent studies also show Saprot's impressive results in B-cell conformational epitope prediction [19], protein engineering [42] and drug-target interaction [28], etc. These results show that Saprot has an exceptional 'one-model-fits-all' capability (Fig. 1c, d and Supplementary Table 5), critical to achieve SaprotHub's community collaboration vision.

We developed ColabSaprot by integrating Saprot into Google Colab's infrastructure to support PLM training. ColabSaprot enables seamless deployment and execution of various task-specific re-trained Saprot models, eliminating the need for environment setup and code debugging. It also allows researchers to initiate training sessions with just a few clicks (Fig. 2a). ColabSaprot is designed to accommodate all tasks within the original Saprot framework, enabling direct prediction for tasks such as zero-shot mutation effect prediction [12, 8, 35] and protein sequence design [16, 3]. For mutation effect prediction, it implements single-site, multi-site, and whole protein sequence single-site saturation mutation. For sequence design, it can design sequences based on backbone structures or redesign specific protein regions, such as antibody variable regions, while keeping the scaffold fixed (Methods).

For supervised re-training tasks, users can fine-tune ColabSaprot with their own data. Here, we introduce a parameter-efficient fine-tuning technique by integrating lightweight adapter networks into ColabSaprot (Fig. 2b). During training, only adapter parameters are updated, achieving nearly the same accuracy as fine-tuning all Saprot parameters (Supplementary Fig. 4). This design not only enhances optimization efficiency [41, 63], but more importantly, as proposed here for the first time, it enables collaborative and centralized mechanisms for these biologist fine-tuned, task-specific PLMs (or Saprot) within the community, especially in the cloud environment (Fig. 2c, d, e, f and Methods). With adapters, researchers can also easily store and exchange their re-trained Saprot models in SaprotHub by loading or uploading adapter networks instead of the full pre-trained model. Since adapter networks contain much fewer parameters (around 1% of the whole Saprot model), this method greatly reduces storage, communication, and management burdens

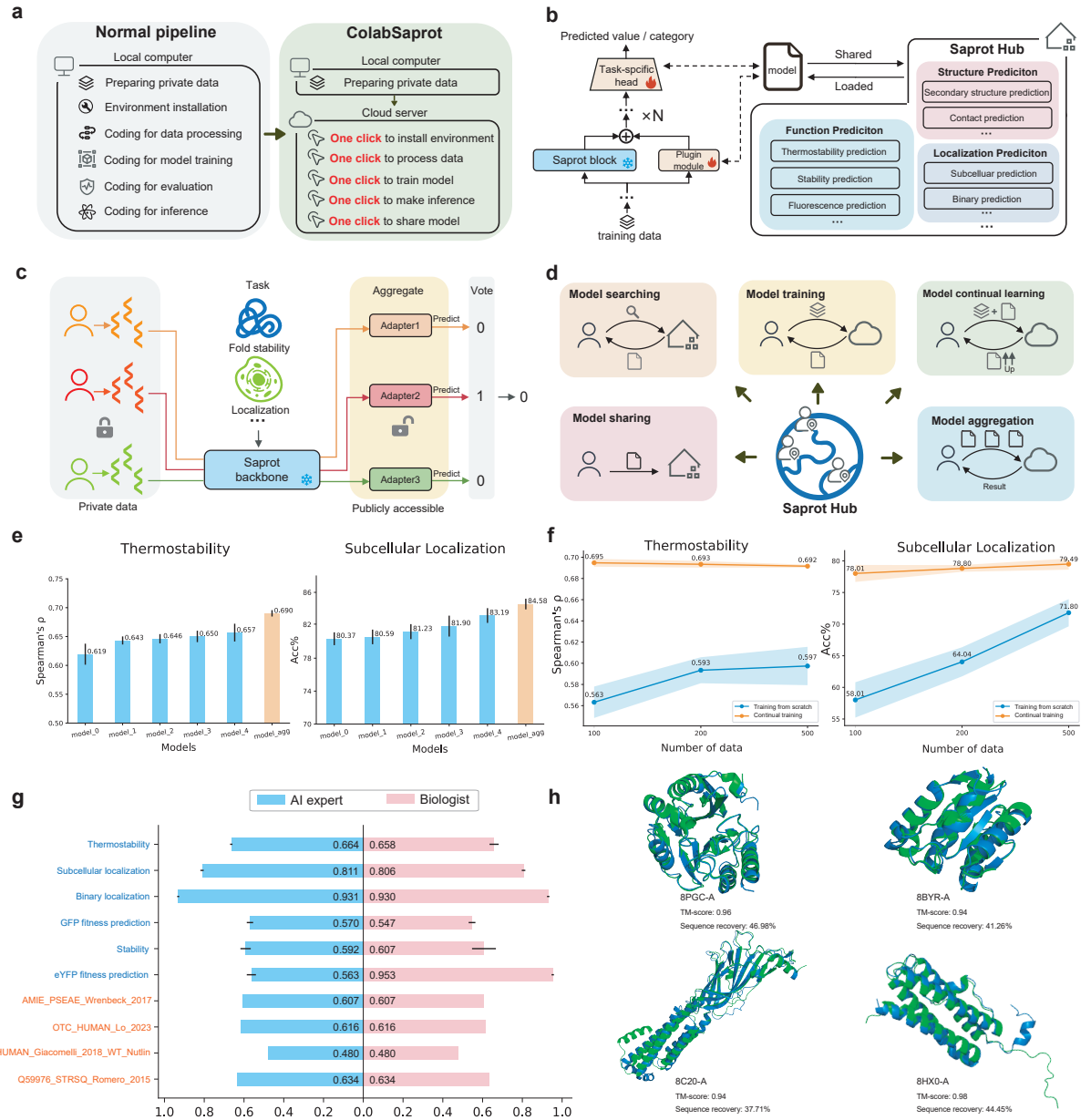


Fig. 2: **Illustration of ColabSaprot and SaprotHub. a,** Comparison of model training and inferring between normal pipeline and ColabSaprot. ColabSaprot streamlines the process, offering simplified procedures for model training and inference with just a few clicks. **b,** A lightweight plugin module (i.e., adapter) is inserted into Saprot for efficient training, sharing and co-construction. **c,** ColabSaprot predicts protein functions by aggregating existing models (or adapters) in SaprotHub without requiring private data. **d,** SaprotHub enables biologists to co-share, co-build, co-use and collaborate within the community. **e,** ColabSaprot's community-wide model collaboration mechanism (see **c**) allows it to achieve higher performance (orange bars) by aggregating individually trained models (blue bars). Each individual model is trained with its own data, which may or may not overlap. **f,** By continually learning on models trained and shared by other biologists, ColabSaprot substantially outperforms those training-from-scratch models, especially when users lack sufficient training data (x-axis represents the number of training examples). **g,** User study on supervised fine-tuning and zero-shot mutation effect prediction tasks. **h,** User study on the inverse folding task. Experimental structures are shown in green, while predicted structures are in blue.

on SaprotHub, making accessibility, co-construction, co-utilization, and co-sharing possible for a broader scientific community (Fig. 2b, c, d).

Additionally, we developed many features for ColabSaprot. Our interface supports nine types of data inputs, including SA sequences, AA sequences, UniProt IDs, and PDB files, etc. We offer automatic conversion of input formats, division of training and testing sets, overfitting checks, automatic best checkpoint saving, evaluation, and visualization. To address GPU memory limitations, we implemented an adaptive batch size feature based on gradient accumulation [22]. We also provide advanced settings that allow researchers to extend code and customize training hyperparameters to suit their specific needs.

Finally, we developed SaprotHub, a centralized model store for seamlessly sharing and co-building peer-retrained Saprot models within the biology community (Fig. 2d). Through the ColabSaprot interface, we implemented features like model storage, sharing, search, continual learning, and aggregation for better performance. This brings two key benefits to SaprotHub: (1) Biologists can share their trained models without compromising sensitive data, overcoming data sharing challenges and enabling the sharing of research outcomes (Fig. 2c); (2) Biologists can continually train models based on shared models from peers, especially when data is limited, as fine-tuning on a superior initialization model typically improves predictive accuracy (Fig. 2f). As more biologists join SaprotHub, it will accumulate a large collection of models for various protein prediction tasks. Additionally, multiple Saprot models will be available for the same protein function. New users can enhance predictive performance (Fig. 2e) by building upon existing models through adapter aggregation (Fig. 2c and Methods) or advanced ensemble learning [21].

SaprotHub has attracted dozens of OPMC developers, sharing over 30 task-specific models. Among these, the Zheng team submitted a high-quality eYFP fluorescence fitness prediction model with a Spearman correlation (ρ) of 0.94, nearing wet lab accuracy on double-site and triple-site mutation tasks (see Methods: SaprotHub page, model name: Model-EYFP_100K-650M). Trained on the largest proprietary fluorescent protein mutation dataset of 100,000 experimentally-tested variants, this model's impressive accuracy is crucial for advancing fluorescent protein design and applications in biotechnology, medical research and evolutionary biology. Additionally, the Chang team recently utilized ColabSaprot to conduct zero-shot single-point mutation predictions on a uracil-N-glycosylase variant, eTDG, and experimentally validated its T-to-G editing efficiency. Among the top 20 predicted mutations, 17 exhibited higher editing efficiency than the wild type, with L74E, H11K, and L74Q showing nearly a 2-fold increase (see Methods and Supplementary Table 6).

We also conducted a user study by recruiting 12 biology researchers (without an ML background) and compared their performance with that of an AI expert (Methods). The results demonstrated that, with ColabSaprot and SaprotHub, biology researchers can train and use state-of-the-art protein models with performance comparable to that of an AI expert (Fig. 2g,h). In some scenarios, such as the eYFP fitness prediction task, biologists using existing models in SaprotHub achieved notably higher predictive accuracy than AI experts, particularly when the training dataset was unavailable or limited.

SaprotHub and ColabSaprot empower biology researchers to easily train and share advanced protein ML models for various prediction tasks, even without expertise in ML or coding. This capability enables the entire protein research community to contribute models, fostering collaboration and promoting the dissemination of high-quality, peer-trained models. We have open-sourced the code for both Saprot and ColabSaprot (Methods), allowing other PLMs to adopt a similar approach to build their own PLM Hub or join SaprotHub. Our OPMC members are continuously integrating more PLMs (such as ESM, ProtTrans, etc.) into the OPMC framework, vastly expanding access to a diverse array of PLMs

This community-wide participation approach to protein modeling aligns with the OPMC vision. Our goal here is to inspire and foster the cooperative construction of open protein prediction models through SaprotHub. We envision SaprotHub as the catalyst that initializes OPMC, driving innovation and collaboration in the field.

References

[1] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, pages 1–3, 2024.

[2] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

[3] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[4] Diego del Alamo. The issues when training on the afdb structure data. *twitter: https://x.com/DdelAlamo/status/1795353297580445851*, 2024.

[5] Jurgen Drews. Drug discovery: a historical perspective. *science*, 287(5460):1960–1964, 2000.

[6] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[7] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[8] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

[9] Benoit Gaujac, Jérémie Donà, Liviu Copoiu, Timothy Atkinson, Thomas Pierrot, and Thomas D Barrett. Learning the language of protein structure. *arXiv preprint arXiv:2405.15840*, 2024.

[10] Michael H Glickman and Aaron Ciechanover. The ubiquitin-proteasome proteolytic pathway: destruction for the sake of construction. *Physiological reviews*, 2002.

[11] Thomas Hayes, Roshan Rao, Halil Akin, Nicholas James Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Quy Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul Santiago Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024.

[12] Yan He, Xibin Zhou, Chong Chang, Ge Chen, Weikuan Liu, Geng Li, Xiaoqi Fan, Mingsun Sun, Chensi Miao, Qianyue Huang, et al. Protein language models-assisted optimization of a uracil-n-glycosylase variant enables programmable t-to-g and t-to-c base editing. *Molecular Cell*.

[13] Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Martin Steinegger, and Burkhard Rost. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, pages 2023–07, 2023.

[14] Brian Hie, Ellen D Zhong, Bonnie Berger, and Bryan Bryson. Learning the language of viral evolution and escape. *Science*, 371(6526):284–288, 2021.

[15] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[16] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems*, 35:38873–38884, 2022.

[19] Nikita V Ivanisenko, Tatiana I Shashkova, Andrey Shevtsov, Maria Sindeeva, Dmitriy Umerenkov, and Olga Kardymon. Sema 2.0: web-platform for b-cell conformational epitopes prediction using artificial intelligence. *Nucleic Acids Research*, page gkae386, 2024.

[20] François Jacob and Jacques Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of molecular biology*, 3(3):318–356, 1961.

[21] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*, 2023.

[22] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[23] Melissa J Landrum, Jennifer M Lee, Mark Benson, Garth R Brown, Chen Chao, Shanmuga Chitipiralla, Baoshan Gu, Jennifer Hart, Douglas Hoffman, Wonhee Jang, et al. Clinvar: improving access to variant interpretations and supporting evidence. *Nucleic acids research*, 46(D1):D1062–D1067, 2018.

[24] Mingchen Li, Yang Tan, Xinzhu Ma, Bozitao Zhong, Huiqun Yu, Ziyi Zhou, Wanli Ouyang, Bingxin Zhou, Liang Hong, and Pan Tan. Prosst: Protein language modeling with quantized structure and disentangled attention. *bioRxiv*, pages 2024–04, 2024.

[25] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[26] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*, 2020.

[27] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

[28] Zhaohan Meng, Zaiqiao Meng, and Iadh Ounis. Fusiondti: Fine-grained binding discovery with token-level fusion for drug-target interaction. *arXiv preprint arXiv:2406.01651*, 2024.

[29] Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. Colabfold: making protein folding accessible to all. *Nature methods*, 19(6):679–682, 2022.

[30] Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

[31] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[32] Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Mahdi Pourmirzaei, Farzaneh Esmaili, Mohammadreza Pourmirzaei, Duolin Wang, and Dong Xu. Prot2token: A multi-task framework for protein language processing using autoregressive language modeling. *bioRxiv*, pages 2024–05, 2024.

[34] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[35] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[36] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

[37] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2023.

[38] Kotaro Tsuboyama, Justas Dauparas, Jonathan Chen, Elodie Laine, Yasser Mohseni Behbahani, Jonathan J Weinstein, Niall M Mangan, Sergey Ovchinnikov, and Gabriel J Rocklin. Mega-scale experimental analysis of protein folding stability in biology and design. *Nature*, 620(7973):434–444, 2023.

[39] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pages 2022–02, 2022.

[40] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure

database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[41] Shuai Zeng, Duolin Wang, and Dong Xu. Peft-sp: Parameter-efficient fine-tuning on large protein language models improves signal peptide prediction. *bioRxiv*, pages 2023–11, 2023.

[42] Ziyi Zhou, Liang Zhang, Yuanxi Yu, Banghao Wu, Mingchen Li, Liang Hong, and Pan Tan. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications*, 15(1):5566, 2024.

OPMC (senior) member list

Sergey Ovchinnikov, MIT
Martin Steinegger, Seoul National University
Kevin K. Yang, Microsoft
Michael Heinzinger, Technische Universität München
Pascal Notin, Harvard University
Pranam Chatterjee, Duke University
Jia Zheng, Westlake University
Stan Z. Li, Westlake University
Xing Chang, Westlake University
Huaizong Shen, Westlake University
Noelia Ferruz, The Centre for Genomic Regulation (CRG)
Rohit Singh, Duke University
Debora Marks, Harvard University
Anping Zeng, Westlake University
Jijie Chai, Westlake University
Anthony Gitter, University of Wisconsin-Madison
Anum Glasgow, Columbia University
Milot Mirdita, Seoul National University
Philip M. Kim, University of Toronto
Christopher Snow, Colorado State University
Vasilis Ntranos, University of California
Philip A. Romero, Duke University
Jianyi Yang, Shandong University
Caixia Gao, Chinese Academy of Sciences

OPMC (regular) member list

Jiawei Zhang, Westlake University
Yuyang Tao, ShanghaiTech University
Fengyuan Dai, Westlake University
Xuting Zhang, Westlake University
Yan He, Westlake University

New authors will be added until the final accepted version.

Methods

Constructing SA Protein Sequence

The Structure-Aware (SA) vocabulary encompasses both residue and structure information, as illustrated in Fig. 1**a**. Given a protein P, its primary sequence can be denoted as $(s_1, s_2, ..., s_n)$, where $s_i \in \mathcal{V}$ represents the residue at the i_{th} site, and $\mathcal{V}$ represents residue alphabet. Drawing inspiration from the vector quantization learning technique [64], we encode protein tertiary structures into discrete residue-like structural tokens. Here, we use Foldseek [39], a fast and accurate protein structure aligner. Through Foldseek, we have a structure alphabet $\mathcal{F}$, wherein P is expressed as the sequence $(f_1, f_2, ..., f_n)$, with $f_j \in \mathcal{F}$ representing the 3Di token for the j_{th} residue site. To maintain simplicity, we adhere to the default configuration of Foldseek, which sets the size m of $\mathcal{F}$ to 20. We then combine the residue and structure tokens per residue site, generating a new structure-aware protein sequence $P = (s_1f_1, s_2f_2, ..., s_nf_n)$, where $s_if_i \in \mathcal{V} \times \mathcal{F}$ is the so-called SA token naturally fusing both residue and geometric conformation information. The SA-token protein sequence can then be fed into a standard Transformer encoder as basic input. It's important to note that we also introduce a mask signal "#" to both residue and structure alphabet, which results in "s_i#" and "#f_i" that indicate only residue or structure information is available. The size of the SA vocabulary is $21 \times 21 = 441$.

Saprot architecture and pre-training

Saprot employs the same network architecture and parameter size as the 650M version of ESM-2 [25], which is inspired by the BERT [47] model in the Natural Language Processing (NLP) field. The main distinction lies in the expanded embedding layer, which encompasses 441 structure-aware tokens instead of the original 20 residue tokens. This nearly identical architecture enables straightforward comparison with the ESM model.

Saprot is pre-trained with the Masked Language Modeling (MLM) objective, similar to ESM-2 and BERT. Formally, for a protein sequence $P = (s_1f_1, s_2f_2, ..., s_nf_n)$, the input and output can be represented as: $input : (s_1f_1, ..., \#f_i, ..., s_nf_n) \rightarrow output : s_if_i$ (Fig. 1b). Given that the 3Di token may not always be accurate for certain regions in predicted structures by AlphaFold2, f_i in $\#f_i$ is made visible during training to reduce the emphasis the model places on its predictions.

Structures predicted by AlphaFold2 (AF2) are accompanied by confidence scores, referred to as pLDDT, which provide an assessment of the precision of atom coordinates. Therefore, we have implemented special handling for regions with low pLDDT scores. During the pre-training process, when regions with pLDDT scores lower than 70 are chosen for MLM prediction, we predict the "s_i#" token. Here, the input SA sequence is masked with the "##" token. This approach forces the model to focus on predicting the types of residues in these regions based on their context. In cases where regions with lower pLDDT scores are not selected for MLM prediction, the input still use the "s_i#" token. This ensures that the model solely relies on residue context (instead of the structural context) in these regions to assist in predicting other tokens. For downstream task prediction phases, we maintain consistency with the training data by handling regions with lower pLDDT scores (still with 70 as threshold). Specifically, tokens within these regions are represented as "s_i#", with only residue tokens being visible.

Processing pre-training data

We followed the procedures outlined in ESM-2 [25] to generate filtered protein data. Subsequently, we acquired all AF2 structures via the AlphaFoldDB website (https://alphafold.ebi.ac.uk/), utilizing the UniProt IDs of protein sequences. Proteins without structures in AlphaFoldDB were omitted for pre-training. This process yielded a collection of approximately 40 million structures. Employing Foldseek, we encoded these structures into 3Di tokens. Subsequently, we formulated structure-aware sequences by combining residue and 3Di tokens, treating them as a single SA token at each position.

Pre-training hyper-parameters

Following ESM-2 and BERT, during training, 15% of the SA tokens in each batch are masked. We replace the SA token s_if_i with the $\#f_i$ token 80% of the time, while 10% of the tokens are replaced with randomly selected tokens, and the other 10% tokens remain unchanged. For the optimization of Saprot, we adopt similar hyper-parameters to those employed in the ESM-2 training phase. Specifically, we employ the AdamW optimizer [53], setting $\beta_1 = 0.9, \beta_2 = 0.98$ and we utilize L_2 weight decay of 0.01. We gradually increase the learning rate from 0 to 4e-4 over the first 2000 steps and linearly lower it to 5e-4 from 150K steps to 1.5M steps. The overall training phase lasts approximately 3M steps. Like the ESM model, we also truncate them to a maximum of 1024 tokens, and our batch size consists of 512 sequences. Additionally, we also employ mixed precision training to train Saprot.

Baseline models

We compared Saprot to several prominent protein language models (see Supplementary Table 2,3). For supervised learning tasks, we compared Saprot with ESM-2 (the 650M version) [25], ProtBert (the BFD version) [6], MIF-ST [67], and GearNet [69]. The first two models utilize residue sequences as input, while the latter two models incorporate both residue and 3D structures as input. ESM-2 (650M) stands out as the primary baseline for comparison, given its similar model architecture, size, and training approach when compared to Saprot. ESM-2 also offers a 15 billion (B) parameter version, which can be challenging to fine-tune even on GPUs with 80G memory. Therefore, we only conducted comparisons with ESM-2 (15B) for zero-shot mutational effect prediction tasks, which can be achieved without the need for fine-tuning.

For the zero-shot mutational effect prediction task, we compare with the state-of-the-art ESM-2, ProtBert, ESM-1v [54], Tranception L (without MSA retrieval) [55], MSA Transformer [59], and EVE [8] models. For the protein inverse folding task, we compare with ProteinMPNN [46] as basline (see Fig. 1).

Zero-shot formula

Previous sequence-based protein language models like the ESM models predict mutational effects using the log odds ratio at the mutated position. The calculation can be formalized as follows:

$$\sum_{t \in T} [logP(x_t = s_t^{mt} | x_{\setminus T}) - logP(x_t = s_t^{wt} | x_{\setminus T})] \tag{1}$$

Here T represents all mutations and $s_t \in \mathcal{V}$ is the residue type for mutant and wild-type sequence. We slightly modify the formula above to adapt to the structure-aware vocabulary, where the probability assigned to each residue corresponds to the summation of tokens encompassing that specific residue type, as shown below:

$$\sum_{t \in T} [log \sum_{f \in \mathcal{F}} P(x_t = s_t^{mt} f | x_{\setminus T}) - log \sum_{f \in \mathcal{F}} P(x_t = s_t^{wt} f | x_{\setminus T})] \tag{2}$$

Here $f \in \mathcal{F}$ is the 3Di token generated by Foldseek and $s_t f \in \mathcal{V} \times \mathcal{F}$ is the SA token in our new vocabulary.

Zero-shot benchmarks

ProteinGym [32] is an extensive set of Deep Mutational Scanning (DMS) assays, enabling thorough comparison among zero-shot predictors. We assess all baselines on the substitution branch of the ProteinGym, utilizing its provided protein structures and following the standard evaluation pipeline from the original paper. For evaluation, We adopt Spearman's rank correlation as our metric.

ClinVar [23] serves as a freely accessible and publicly available repository containing information about human genetic variants and interpretations of their significance to disease. In our analysis, we harness the data sourced from EVE [8], and filter out proteins with length greater than 1024. To enhance the reliability of our data, we opt to consider proteins with labels 1 "Gold Stars" or higher, which indicate higher credibility. Following the methodology employed in EVE, we evaluate models' performance using the AUC metric.

Mega-scale [38] uses cDNA display proteolysis to measure the thermodynamic folding stability of protein domains. Its released dataset contains all single mutations and selected double mutants of natural and *de novo* designed proteins. We employ Spearman's rank correlation as our metric.

For each mutation dataset, we provide all variants with the wild-type structure, as AF2 does not reliably distinguish the structural changes induced by single mutations. The ClinVar dataset only provides UniProt IDs, so we manually downloaded all AF2 structures and eliminated proteins without structures in AlphaFoldDB. Both ProteinGym and Mega-scale datasets provide protein structures, either predicted from AF2 or derived from *de novo* design.

Supervised fine-tuning benchmarks

Fine-tuning Saprot with AF2 structure. For benchmarks lacking PDB structures, we retrieve all AF2-predicted structures using UniProt IDs. These benchmarks encompass the Thermostability task from FLIP [45] and the Localization Prediction task from DeepLoc [43]. DeepLoc consists of two branches for subcellular localization prediction: one involving 10 location categories and the other involving binary localization prediction with 2 location categories. We assess the models' performance on both branches.

Fine-tuning Saprot with PDB structure. These tasks provide experimentally determined structures as training data. We evaluate Metal Ion Binding task [50] and a series of tasks from ProteinShake [52], including Structure Class Prediction, Structural Similarity Prediction and Binding Site Detection.

Fine-tuning Saprot without structure. Saprot is specifically developed to incorporate protein structures as valuable input. However, it can still work even in scenarios where structures are not provided. In such cases, the 3Di token is masked within the SA sequence. We evaluate the Fluorescence prediction and Stability prediction tasks from the TAPE [58] benchmark, the AAV dataset from the FLIP [45] benchmark

and β-lactamase landscape prediction from the PEER [66] benchmark. For a given wild type protein, its variants do not offer additional structures.

Dataset splits

In previous literature, a common practice was to partition datasets based on sequence identity. However, a recent benchmark study, ProteinShake [52], mentioned that protein structures exhibit greater conservation than protein sequences. They argue that dissimilar sequences in training and test sets may share similar structures, leading to data leakage issues. Consequently, we adopt a more strict data splitting approach proposed by ProteinShake and split the data based on structure similarity. For all datasets with structures, we employ a 70% structure similarity threshold for splitting. For those without provided structures, we still follow the original data splits.

Supervised fine-tuning hyper-parameters

We employ the AdamW [53] optimizer during fine-tuning, setting $\beta_1 = 0.9, \beta_2 = 0.98$, along with L_2 weight decay of 0.01. We use a batch size of 64 for all datasets. We empirically found that the optimal learning rate for most baselines are in the range of 1e-5 to 5e-5. For **Training with AF2 structure** and **Training with PDB structure**, the optimal learning rate was set to 5e-5, whereas for **Training without structure**, it was set to 1e-5. We fine-tuned all model parameters until convergence and selected the best checkpoints based on their performance on the validation set.

Protein inverse folding

Given the 3D coordinates of a protein backbone, protein inverse folding aims to predict residue sequences that fold into this shape. This exhibits promising applications in protein *de novo* design. While not particularly targeted on this task, Saprot can predict reliable residue sequences from the backbone due to the large-scale pre-training tailored for predicting masked residues based on sequence and structure contexts. To utilize Saprot for inverse folding, we first encode protein backbone into 3Di tokens $(f_1, f_2, ..., f_n)$. Then we add masks to all residue parts of SA tokens, forming a SA sequence $(\#f_1, \#f_2, ..., \#f_n)$, which serves as input for Saprot to predict residue distributions at all positions. In contrast to ProteinMPNN [46] that generates residues in the auto-regressive manner, i.e. generating next token conditioned on all previous outputs, Saprot is able to simultaneously predict all residues with only one forward propagation. As shown in Fig. 1e, in CATH [56] test set, Saprot predicts 16 times faster than ProteinMPNN, while achieves comparable prediction accuracy, which demonstrates its huge potential in protein *de novo* design.

Embedding visualization

We employ t-SNE [65] to visualize the protein representations generated by Saprot and ESM-2 in three datasets (Supplementary Fig. 3). We use the non-redundant version ($PIDE < 40\%$) of the SCOPe [44] database, visualizing all alpha and beta proteins. We additionally visualize proteins in Subcellular Localization and Binary Localization datasets from DeepLoc [43]. To generate protein representations, we adopt the average pooling on the embeddings of the last layer for both Saprot and ESM-2.

Adapter learning

The adapter learning technique originated in the field of NLP [49, 48, 68] and has recently garnered attention in protein-related research. However, these recent preliminary preprints primarily focused on its ability in model accuracy or efficient GPU memory consumption [63, 62, 61, 60]. Through the application of adapters, researchers have observed their ability to alleviate overfitting issues or integrate domain-specific knowledge into protein language models. In this paper, we innovatively integrate it with Google Colaboratory and endow it with deeper capabilities, namely, to promote model sharing and co-construction within the biologist community in the online SaprotHub. This represents a novel perspective on the application of adapter learning within protein biology. In the broader AI community, various types of adapters exist, including Houlsby [49], Pfeiffer [57], Compacter [51], and LoRA [17]. We choose LoRA for its capacity to deliver comparable results with fewer parameters. By integrating learnable low-rank matrices into each Saprot block while freezing the backbone, LoRA enables parameter-efficient fine-tuning and model sharing for these downstream tasks.

Community-wide collaboration

As the SaprotHub community expands, an increasing number of Saprot models will become available for the same protein function. This paves the way for collaborative co-construction among biologists. To illustrate this advantage (Fig. 2e), we employ intuitive approach. In regression tasks, the final prediction is obtained by calculating the mean value of outputs from all models. For classification tasks, we aggregate predicted

categories from all models and select the category with the highest prediction as the final outcome. Note that all models shown in Fig. 2e were run five times using different random seeds. The lightweight adapter technique plays a vital role in model aggregation, addressing the impracticality of loading and sharing multiple large pre-trained models.

Furthermore, more advanced functionalities can be developed. For instance, by injecting multiple adapters into Saprot via SaprotHub, continual learning can be conducted to further enhance performance. Adapters eliminate the need to modify the weights of the original Saprot model, effectively mitigating the issue of catastrophic forgetting. This study introduces a pioneering concept of integrating community wisdom by utilizing shared adapters, thereby ensuring data privacy. This design enables the realization of the OPMC vision.

ColabSaprot notebooks

ColabSaprot comprises three key components. The first component focuses on training and inference, enabling researchers to swiftly set up the runtime environment, preprocess training data, and fine-tune Saprot with ease. We offer advanced hyper-parameter settings, allowing researchers to customize their training strategy based on specific requirements, such as batch size, learning rate, and training steps. After completing the training process, researchers can share the model weights on the SaprotHub community store, benefiting other biologists.

Furthermore, researchers without their own training data can directly access a range of fine-tuned Saprot models from SaprotHub and perform inference for desired protein functions. These existing models also support continual training with additional data, leading to improved prediction accuracy.

The second component focuses on the zero-shot mutational effect prediction task, enabling users to leverage Saprot for predicting fitness changes resulting from residue mutations. We provide multiple options for users to predict single mutations, multi-site mutations, or saturation mutations on given proteins.

The final component of Saprot offers support for the protein inverse folding task based on the backbone structure. Saprot achieves a 16 times faster inference speed than ProteinMPNN while maintaining competitive prediction performance. This capability empowers users to conduct precise protein designs on a large scale.

User study

We recruited 12 participants to use ColabSaprot for 8 protein prediction tasks, including supervised fine-tuning, zero-shot mutation effect prediction, and protein inverse folding. All these participants had an education background in biology but no experience with machine learning. They were required to complete their assigned tasks within 3 days. In comparison, the AI expert is a third-year PhD student in machine learning with over 2 years of research experience in AI for protein studies. Our AI expert directly used the Saprot code from our GitHub repository to perform these tasks. The AI expert conducted these experiments five times using different random seeds with optimal hyper-parameters.

For the supervised fine-tuning, we utilized five publicly available datasets dataset, including those for thermostability prediction, (subcellular and binary) localization prediction, GFP fluorescence prediction, and stability prediction. Additionally, we introduced a proprietary eYFP fitness prediction dataset collected by Zheng's lab, which has 3087 and 3088 samples for validation and testing. To reduce training time and computing power consumption, we randomly selected 1,000 samples from the training set of each dataset while keeping the validation and test sets unchanged.

Among the 12 participants, 10 conducted the supervised fine-tuning tasks. Each of these participants was assigned three tasks from a set of six. In other words, each training task was performed by 5 participants. We evaluated their average accuracy.

These participants were required to train the models and provide predictions on the corresponding test sets for subsequent evaluation. It is important to note that for the eYFP task, biology participants were required to perform continual learning on a model that had previously been trained and shared on the SaprotHub platform (see https://huggingface.co/SaProtHub/Model-EYFP-650M), while the AI expert was required to use the base Saprot model after pre-training. The purpose here is to demonstrate the advantage of SaprotHub for researchers who lack training data. For all other tasks, biology participants and the AI expert trained their models using the same base Saprot model (i.e., https://huggingface.co/westlake-repl/SaProt_650M_AF2) for fair evaluation.

For the zero-shot mutational effect prediction, we randomly selected four mutation datasets from ProteinGym benchmark [32] for evaluation. Three of these datasets focus on the impact of mutations on enzyme activity, while the fourth addresses drug resistance. We assigned one participant to conduct predictions on these datasets using ColabSaprot in a zero-shot manner.

For the protein inverse folding task, we assigned one participant to use ColabSaprot to generate protein sequences based on given structures. Subsequently, the participant employed ESMFold (an interface provided by ColabSaprot) to predict the structures of the generated sequences. To assess Saprot's ability on new structures, we selected these recently released structures (see Fig. 2h).

Please found the user study materials at *Data availability* section below.

Wet-lab experimental validation on eTDG

The Chang team utilized ColabSaprot to perform zero-shot mutation effect prediction on a variant of uracil-N-glycosylase called eTDG. They then conducted experimental validation following the procedure outlined in [12] to verify the top 20 predicted mutations by ColabSaprot. Supplementary Table 6 provides the predicted scores and experimentally validated editing efficiencies. Notably, the original paper by [12] reported only 8 of the variants predicted by ColabSaprot.

Data availability

The pre-training dataset for training Saprot is available at https://huggingface.co/datasets/westlake-repl/AF2_UniRef50. Downstream task datasets are all stored at https://huggingface.co/SaProtHub. Data for user study analysis is available at https://drive.google.com/file/d/1LdGRnwt2lttnszNBAJ0F967A8rguPq8b/view?usp=sharing.

Code and service availability

Saprot is an open-sourced model with MIT license. The code is available at https://github.com/westlake-repl/Saprot. The code implementation of ColabSaprot notebook is available at https://github.com/westlake-repl/SaProtHub. ColabSaprot service is available at https://colab.research.google.com/github/westlake-repl/SaprotHub/blob/main/colab/SaprotHub.ipynb. Fine-tuned models can be obtained through SaprotHub https://huggingface.co/SaProtHub.

References

[43] José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017.

[44] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic Acids Research*, 47(D1):D475–D481, 11 2018.

[45] Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11, 2021.

[46] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[47] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[48] Junchen Fu, Fajie Yuan, Yu Song, Zheng Yuan, Mingyue Cheng, Shenghui Cheng, Jiaqi Zhang, Jie Wang, and Yunzhu Pan. Exploring adapter-based transfer learning for recommender systems: Empirical studies and practical insights. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 208–217, 2024.

[49] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR, 2019.

[50] Mingyang Hu, Fajie Yuan, Kevin Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. Exploring evolution-aware & -free protein language models as protein function predictors. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 38873–38884. Curran Associates, Inc., 2022.

[51] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021.

[52] Tim Kucera, Carlos Oliver, Dexiong Chen, and Karsten Borgwardt. Proteinshake: Building datasets and benchmarks for deep learning on protein structures. *Advances in Neural Information Processing Systems*, 36, 2024.

[53] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017.

[54] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in neural information processing systems*, 34:29287–29303, 2021.

[55] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pages 16990–17017. PMLR, 2022.

[56] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

[57] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*, 2020.

[58] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

[59] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[60] Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *bioRxiv*, pages 2023–12, 2023.

[61] Amelie Schreiber. Esmbind and qbind: Lora, qlora, and esm-2 for predicting binding sites and post translational modification. *bioRxiv*, pages 2023–11, 2023.

[62] Samuel Sledzieski, Meghana Kshirsagar, Minkyung Baek, Bonnie Berger, Rahul Dodhia, and Juan Lavista Ferres. Democratizing protein language models with parameter-efficient fine-tuning. *bioRxiv*, 2023.

[63] Samuel Sledzieski, Meghana Kshirsagar, Bonnie Berger, Rahul Dodhia, and Juan Lavista Ferres. Parameter-efficient fine-tuning of protein language models improves prediction of protein-protein interactions.

[64] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[65] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[66] Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Ma Chang, Runcheng Liu, and Jian Tang. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35:35156–35173, 2022.

[67] Kevin K Yang, Niccolò Zanichelli, and Hugh Yeh. Masked inverse folding with sequence transfer for protein representation learning. *Protein Engineering, Design and Selection*, 36:gzad015, 2023.

[68] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1469–1478, 2020.

[69] Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. *arXiv preprint arXiv:2203.06125*, 2022.

Acknowledgements

We thank Nan Li and Westlake University HPC Center for computing resources, and Jing Huang, Jianyang Zeng, Dangshen Li, Longxing Cao, Dapeng Li for discussions and paper advice. We thank John M. Jumper for providing insights about AF2 Evoformer and for his advice on the SA token design via email. This work is supported by the National Key Research and Development Program of China (No. 2022ZD0115100), the National Natural Science Foundation of China (No. U21A20427), the Westlake Center of Synthetic Biology and Integrated Bioengineering (WE-SynBio), and the Research Center for lndustries of the Future (No.WU2022C030).

Author contributions

F.Y. conceived and led this research. J.S. performed the main research and managed technical implementation. J.S., F.Y., and J.S. (Junjie) designed the SA token and Saprot. Z.L. and J.S. developed the ColabSaprot and SaprotHub. F.Y., J.S., and S.O. designed ColabSaprot and SaprotHub. S.O. and M.S. engaged in discussions of ColabSaprot and SaprotHub, provided many constructive suggestions, and revised the manuscript. C.H. conducted partial zero-shot mutation experiments. Y.Z. collected the AFDB dataset. Y.H., X.C. and X.Z. conducted experiments on eTDG. D.M. participated in early wet lab experiments. J.S. and F.Y. wrote the manuscript. OPMC authors contributed as described in the paper, including manuscript proofreading, idea discussions, expert advice, promoting, testing and development of ColabSaprot.

Supplementary information

OPMC

The Open Protein Modeling Consortium (OPMC) is a collaborative initiative designed to unify the efforts of the protein research community. Its mission is to facilitate the sharing and co-construction of resources, with a particular focus on individually trained decentralized models, thereby advancing protein modeling through collective contributions. OPMC offers a platform that supports a wide range of protein function predictions, aiming to make advanced protein modeling accessible to researchers irrespective of their machine learning expertise.

Here are some Q&A related to OPMC that we have gathered from our steering committee members and other researchers: https://theopmc.github.io/

Be an OPMC member

The complete list of OPMC authors will be finalized in the last version of the paper. OPMC encompasses two membership categories: the steering committee and regular members.

Steering Committee: The steering committee comprises esteemed researchers with doctoral degrees who assume advisory roles within OPMC. Their primary responsibilities include:

- Providing constructive suggestions for OPMC and SaprotHub.
- Supporting academic activities such as organizing workshops, tutorials, and participating in peer review processes.
- Encouraging direct involvement of team members in the development of SaprotHub or OPMC.
- Participating in the review process for new members joining OPMC.
- Contributing high-quality open-source datasets and models to enhance OPMC's resources.
- Making direct contributions to this paper.
- Making other contributions or providing academic services that drive the progress of OPMC.

Regular Members: Regular members actively contribute to OPMC through practical means. This category includes independent researchers and professionals who:

- Participate in research, development, and collaborative efforts to advance OPMC's goals and objectives.
- Create innovative adapters for SaprotHub, trained on either private or publicly available data.
- Contributing more high-quality datasets to enrich the available resources.
- Aid in the development and administration of SaprotHub or ColabSaprot.

Membership Eligibility: We enthusiastically welcome individuals who make substantial contributions to OPMC and SaprotHub. The eligibility of regular members will be determined by the steering committee. If you are interested in becoming a member of OPMC, please reach out to us. Note that all OPMC members will be included as authors in the final version of this paper. The final list of members is expected to be finalized within the next 4 to 12 months.

Dataset	Evaluation Metric	Train	Valid	Test
Training with AF2 structure				
Thermostability [45]	Spearman's ρ	5310	706	706
Subcellular localization [43]	Acc	10414	1368	1368
Binary localization [43]	Acc	6707	698	807
Training with PDB structure				
Structural similarity [52]	Spearman's ρ	300699	4559	4850
Structure class [52]	ACC	7990	955	1005
Metal ion binding [50]	ACC	5797	719	719
Binding site detection [52]	Mcc	2368	442	464
Training without structure				
Fluorescence [58]	Spearman's ρ	20963	5235	25517
Stability [58]	Spearman's ρ	53614	2512	12851
AAV [45]	Spearman's ρ	22246	2462	50432
β**-lactamase** [66]	Spearman's ρ	4158	520	520

Supplementary Table 1: Downstream dataset descriptions after all data pre-processing. All datasets that require structures as input are split based on structure similarity with 70 as threshold. For **Training with AF2 structure**, proteins with structures not available in AlphaFoldDB were excluded from both training and testing evaluations.

Model	ProteinGym	Mega-scale	ClinVar
	Spearman's ρ	Spearman's ρ	AUC
ESM-2 (650M) [25] (Science, 2023)	0.414	0.478	0.862
ESM-2 (15B) [25] (Science, 2023)	0.401	0.467	0.843
ProtBert [6] (TPAMI, 2021)	0.350	0.341	0.862
ESM-1v [54] (NeurIPS, 2021)	0.374	0.374	0.891
Tranception L [55] (ICML, 2022)	0.374	0.307	0.845
MSA Transformer [59] (ICML, 2021)	0.421	0.423	0.854
EVE [8] (Nature, 2021)	0.433	0.260	0.878
Saprot	**0.457**	**0.574**	**0.909**

Supplementary Table 2: More baseline results on zero-shot mutational effect prediction.

Task	Category	Metric	ESM-2	ProtBert	MIF-ST [67]	GearNet[69]	**Saprot**
Thermostability	Training w/ AF2 structure	Spearman's ρ	0.673	0.621	0.678	0.454	0.697
Subcellular localization		Acc%	84.67	78.65	82.46	75.73	86.18
Binary localization		Acc%	93.56	93.31	92.08	90.72	94.67
Structure class	Training w/ PDB structure	Acc%	79.90	71.57	71.63	74.52	80.80
Structural similarity		Spearman's ρ	0.677	0.601	0.555	0.101	0.695
Binding site detection		Mcc	0.540	0.459	0.504	0.415	0.602
Metal Ion binding		Acc%	74.97	68.19	69.58	67.61	79.97
Fluorescence	Training w/o structure	Spearman's ρ	0.698	0.698	-	-	0.699
Stability		Spearman's ρ	0.780	0.753	-	-	0.779
β-lactamase		Spearman's ρ	0.904	0.881	-	-	0.890
AAV		Spearman's ρ	0.813	0.783	-	-	0.829

Supplementary Table 3: More baseline results on supervised learning tasks. MIF-ST and GearNet are not applicable when protein structures are not available as input.

Task / Dataset	**Metric**	ProstT5	**Saprot**
ClinVar	AUC	0.620	**0.909**
Mega-scale	Spearman's ρ	0.194	**0.574**
Subcellular localization	Acc%	83.26	**86.18**
Binary localization	Acc%	94.11	**94.67**
Protein inverse folding	pLDDT	76.24	**77.96**
	RMSD	2.28	**2.07**
	TM-score	0.78	**0.82**
	Sequence recovery	33.72	**36.15**

Supplementary Table 4: Comparison between ProstT5 and Saprot in several tasks. ProstT5 is a seq2seq T5 model where the input, if it is an amino acid sequence, corresponds to the output of the 3Di sequence, and vice versa. In general, Saprot is comparable to ProstT5 in the protein inverse folding task, but substantially outperform it on other tasks, e.g., zero-shot mutational effect prediction and supervised learning tasks.

Supported Tasks	
Protein-level Classification	Active Site Prediction
Fold Class Prediction	Post-Translational Modification (PTM) Site Prediction
Localization Prediction	Mutation Impact Prediction
Subcellular Localization	Transmembrane Region Prediction
Domain Classification	Conservation Score Prediction
Membrane Protein Classification	Interface Residue Prediction
Enzyme Classification	Enzyme Binding Site Prediction
Post-translational Modification Site Classification	Metal Binding Site Prediction
Protein Stability Classification	Disease-Associated Variant Prediction
Protein Solubility Classification	Immune Receptor Binding Prediction
Antigenicity Classification	Glycosylation Site Prediction
Protein Aggregation Classification	**Protein-protein Classification**
Protein Family Classification	Protein-Protein Interaction (PPI) Prediction
Protein Structural Class Prediction	Interaction Type Classification
Protein Subunit Interface Classification	Disease-Associated Interaction Prediction
Protein Motif Classification	Mutational Impact Classification
Protein Epitope Classification	Functional Annotation Classification
Protein Interaction Network Classification	Interaction Network Module Detection
Protein Toxicity Classification	Interaction Stability Prediction
Protein Expression Level Classification	**Protien-protein Regression**
Protein Dynamics Classification	Interaction Strength Prediction
Protein Flexibility Classification	Binding Free Energy Calculation
Protein Function Annotation	Interaction Affinity Prediction
Protein Interaction Type Classification	Mutational Impact Quantification
Protein Disease Association Classification	Interaction Kinetics Prediction
Protein Functional Domain Classification	**Zero-shot mutational Effect prediction**
Protein-level Regression	Enzyme Activity Prediction
Thermal Stability Prediction	Virus Fitness Prediction
Fluorescence Intensity Prediction	Driver Mutation Prediction
Binding Affinity Prediction	Protein Stability Prediction
Solubility Prediction	Immune Escape Prediction
Expression Level Prediction	MHC Binding Prediction
Enzymatic Activity Prediction	Pathogenicity Prediction
Hydrophobicity Prediction	Membrane Protein Function Prediction
Protease Sensitivity Prediction	Functional Domain Impact Prediction
Protein Interaction Strength Prediction	**Protein Sequence Design**
Drug Binding Affinity Prediction	Enzyme Function Optimization
Protein Stability Change upon Mutation Prediction	Protein Stability Enhancement
Protein Ligand Binding Kinetics Prediction	Protein Folding Prediction
Protein Stability in Different Temperatures Prediction	Antibody Design
Protein Stability in Different Solvents Prediction	Development of Novel Protein Drugs
Residue-level Classification	Vaccine Design
Secondary Structure Prediction	Protein Engineering
Binding Site Prediction	Synthetic Antibody Fragments Design

Supplementary Table 5: Tasks that Saprot potentially supports. As a near-universal protein language model, Saprot potentially supports hundreds of protein prediction tasks (also see Fig. 1 c,d). These tasks include, but are not limited to: protein-level classification, protein-level regression, residue-level classification, protein-protein classification, protein-protein regression, zero-shot mutational effect prediction, and protein sequence design.

Mutation	Predicted score	T>G% (wildtype=10.3865)	Enhancement factor
I37P	4.8993	15.7850	1.5198
L74A	4.7470	12.9784	1.2495
R8M	3.8026	14.7522	1.4203
L74E	3.6393	**20.4948**	**1.9732**
S5M	3.6230	15.0282	1.4469
P43K	3.3585	14.7687	1.4219
P6M	3.3499	15.5822	1.5002
P43R	3.2832	13.6750	1.3166
I37S	3.2649	16.4798	1.5866
I103E	3.2008	11.9370	1.1493
H11P	3.1735	14.6748	1.4129
H11K	3.1643	**18.0260**	**1.7355**
I37R	3.1442	14.8948	1.4341
I103D	3.1406	6.9425	0.6684
F1M	3.0996	6.0477	0.5823
L74Q	3.0893	**19.1058**	**1.8395**
H11A	2.9147	17.1734	1.6534
P4M	2.7864	15.2689	1.4701
I37A	2.7000	16.3456	1.5737
K252L	2.6721	1.2497	0.1203

Supplementary Table 6: The editing efficiency of T-to-G mutations in the top 20 variants predicted by ColabSaprot. Among the top 20 variants, 17 mutations exhibited higher editing efficiency compared to the wild-type. Specifically, the editing efficiency at the L47E, L74Q and H11K sites were nearly doubled compared to the wild-type.

Benchmark Scores

https://proteingym.org/benchmarks

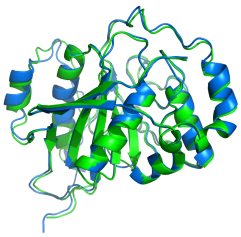
Rank▲	Model name	Model type	Avg. Spearman
1	SaProt (650M)	Protein language model	0.457
2	TranceptEVE L	Hybrid - Alignment & PLM	0.456
3	TranceptEVE M	Hybrid - Alignment & PLM	0.455
4	GEMME	Alignment-based model	0.455
5	TranceptEVE S	Hybrid - Alignment & PLM	0.452
6	ProtSSN (ensemble)	Hybrid - Structure & PLM	0.449
7	ProtSSN (k=20 h=1280)	Hybrid - Structure & PLM	0.442
8	ProtSSN (k=20 h=512)	Hybrid - Structure & PLM	0.441
9	ProtSSN (k=20 h=768)	Hybrid - Structure & PLM	0.44
10	EVE (ensemble)	Alignment-based model	0.439

Rank▲	Model name	Model type	Avg. Spearman
11	ProtSSN (k=30 h=512)	Hybrid - Structure & PLM	0.438
12	ProtSSN (k=30 h=768)	Hybrid - Structure & PLM	0.438
13	ProtSSN (k=30 h=1280)	Hybrid - Structure & PLM	0.438
14	VESPA	Protein language model	0.436
15	Tranception L	Hybrid - Alignment & PLM	0.434
16	MSA Transformer (ensemble)	Hybrid - Alignment & PLM	0.434
17	ProtSSN (k=10 h=1280)	Hybrid - Structure & PLM	0.433
18	EVE (single)	Alignment-based model	0.433
19	ProtSSN (k=10 h=512)	Hybrid - Structure & PLM	0.43
20	Tranception M	Hybrid - Alignment & PLM	0.427

Rank▲	Model name	Model type	Avg. Spearman
21	ProtSSN (k=10 h=768)	Hybrid - Structure & PLM	0.423
22	ESM-IF1	Inverse folding model	0.422
23	MSA Transformer (single)	Hybrid - Alignment & PLM	0.421
24	DeepSequence (ensemble)	Alignment-based model	0.419
25	Tranception S	Hybrid - Alignment & PLM	0.418
26	ESM2 (650M)	Protein language model	0.414
27	DeepSequence (single)	Alignment-based model	0.407
28	ESM-1v (ensemble)	Protein language model	0.407
29	SaProt (35M)	Protein language model	0.406
30	ESM2 (3B)	Protein language model	0.406

Rank▲	Model name	Model type	Avg. Spearman
31	ESM2 (15B)	Protein language model	0.401
32	MIF-ST	Hybrid - Structure & PLM	0.4
33	EVmutation	Alignment-based model	0.395
34	ESM-1b	Protein language model	0.394
35	VESPAl	Protein language model	0.394
36	Progen2 XL	Protein language model	0.391
37	ESM2 (150M)	Protein language model	0.387
38	MIF	Inverse folding model	0.383
39	Progen2 L	Protein language model	0.38
40	Progen2 M	Protein language model	0.379

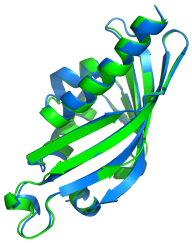
Supplementary Fig. 1: Saprot (checkpoint: https://huggingface.co/westlake-repl/SaProt_650M_AF2) ranked first on the public ProteinGym leaderboard until May 2024, surpassing over 60 well-known methods. Note that Saprot is a single model, while most other top-ranked models are hybrid models.



1FGX-A

TM-score: 0.92

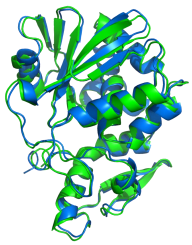
Sequence recovery: 56.62%



4GB5-A

TM-score: 0.98

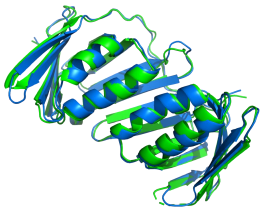
Sequence recovery: 39.19%



1NIO-A

TM-score: 0.97

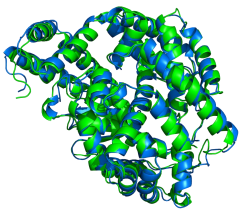
Sequence recovery: 41.30%



3P91-A

TM-score: 0.96

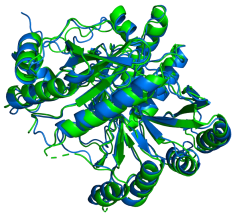
Sequence recovery: 36.73%



1KA2-A

TM-score: 0.98

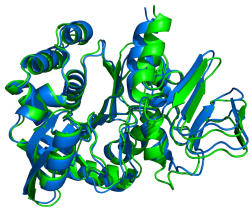
Sequence recovery: 44.45%



1AOP-A

TM-score: 0.96

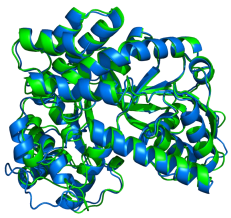
Sequence recovery: 47.81%



2CU2-A

TM-score: 0.96

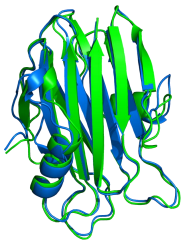
Sequence recovery: 52.54%



3FEF-A

TM-score: 0.95

Sequence recovery: 38.02%



1GWY-A

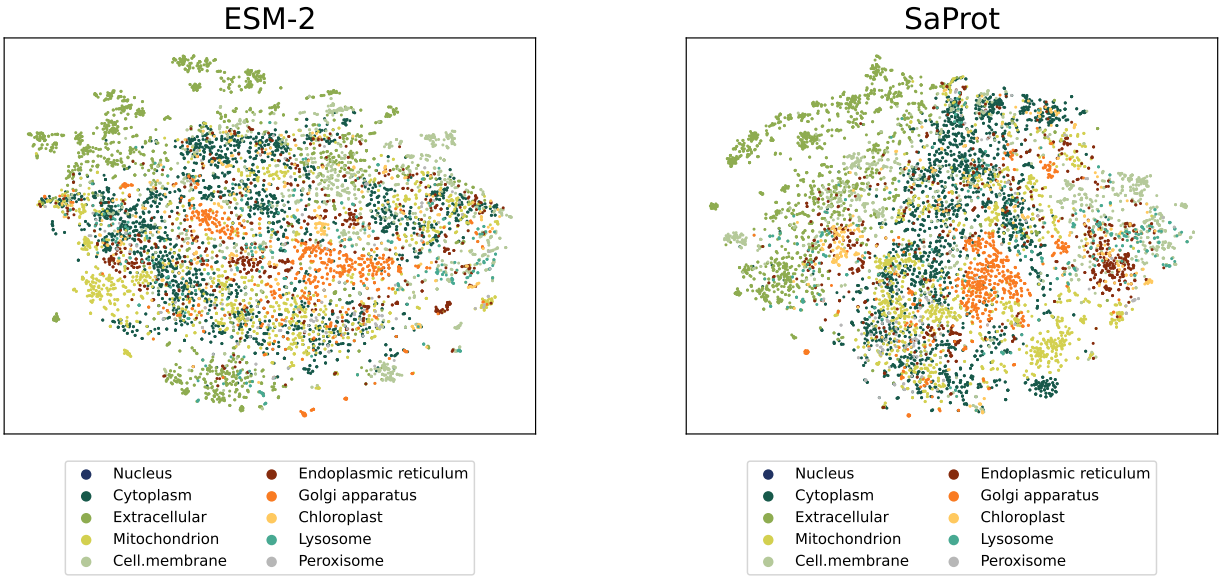
TM-score: 0.94

Sequence recovery: 37.71%

Supplementary Fig. 2: Case studies of protein inverse folding using Saprot. The blue-colored protein structures are predicted using ESMFold, with residue sequences generated by Saprot, and the green-colored structures are the experimentally determined structures. Given the backbone, Saprot is capable of predicting diverse sequences with few residue overlap while maintaining almost the same 3D structure.

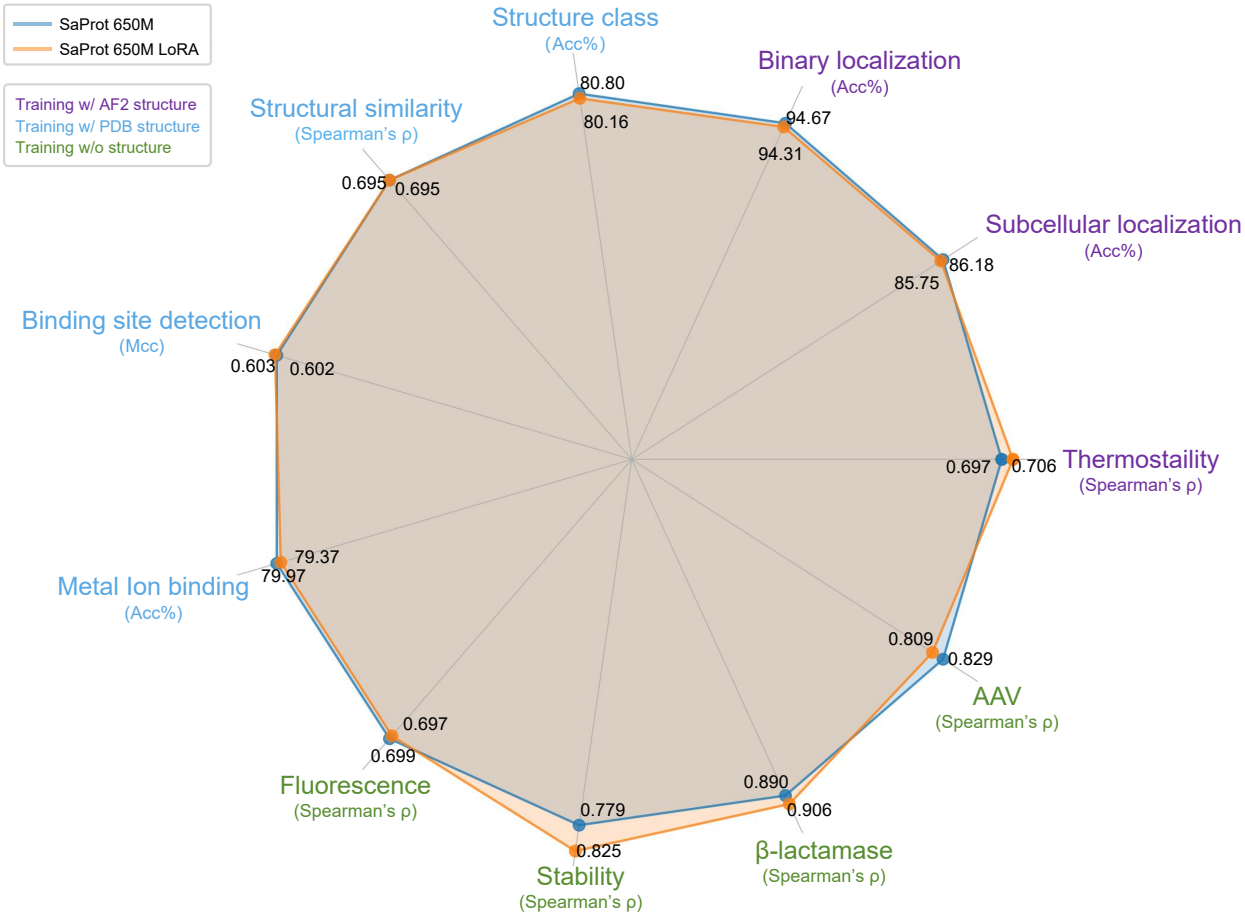
(a) Embedding visualizations of ESM-2 650M version and Saprot 650M version on SCOPe database. We use the non-redundant version ($PIDE < 40\%$) of the SCOPe database.

(b) Embedding visualizations of ESM-2 and Saprot on binary localization dataset.



(c) Embedding visualizations of ESM-2 and Saprot on subcellular localization dataset.

Supplementary Fig. 3: Embedding visualization



Supplementary Fig. 4: Comparison between full model fine-tuning and parameter efficient fine-tuning. While greatly reducing trainable parameters, fine-tuning Saprot using adapter module achieves competitive performance on all experimental settings, which demonstrates its applicability on various biological tasks.